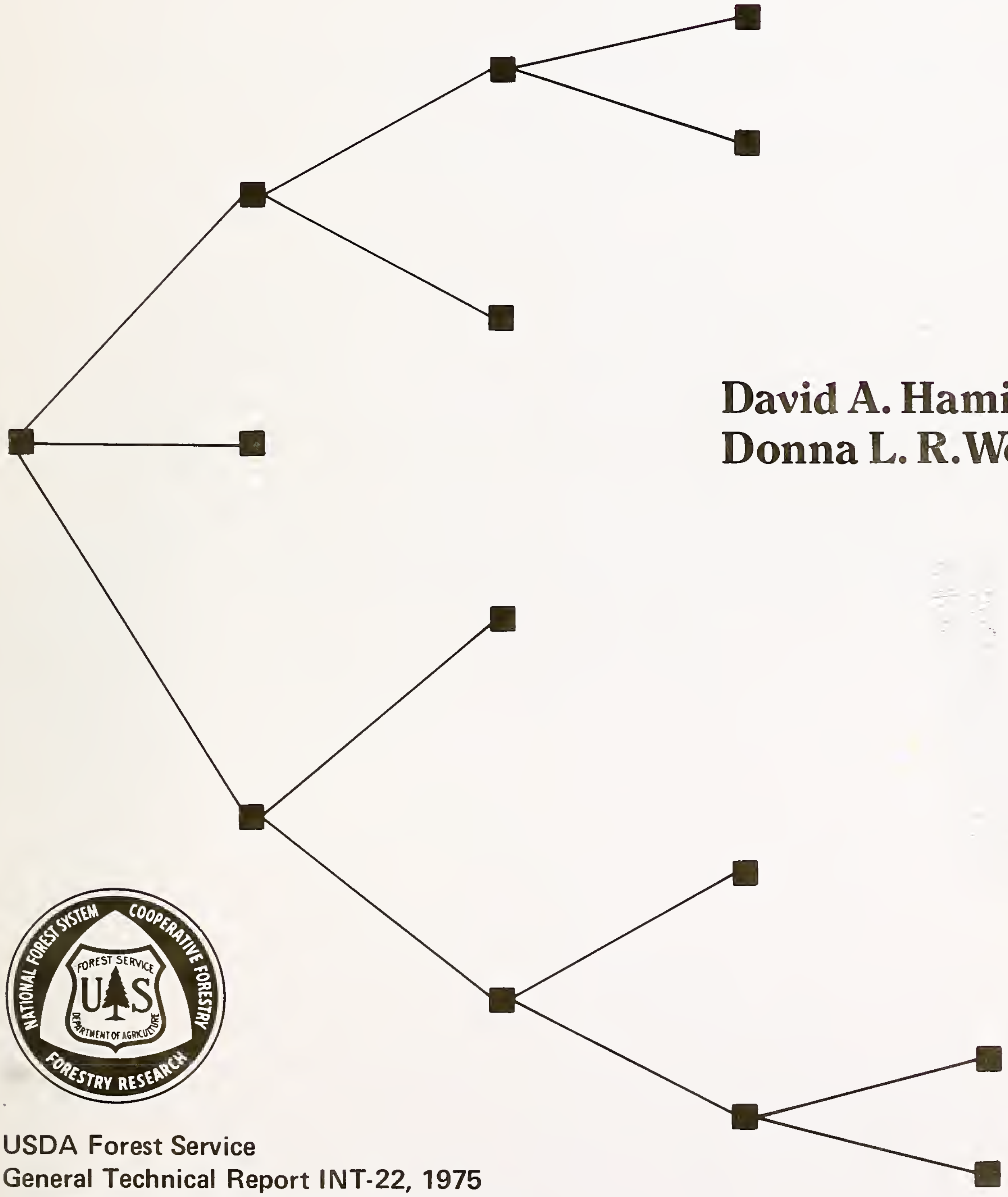
SCREEN: a computer program to identify predictors of dichotomous dependent variables



David A. Hamilton, Jr., and
Donna L. R. Wendt

USDA Forest Service
General Technical Report INT-22, 1975
INTERMOUNTAIN FOREST AND RANGE
EXPERIMENT STATION
Ogden, Utah 84401

USDA Forest Service
General Technical Report INT-22
July 1975

SCREEN: a computer program to identify predictors of dichotomous dependent variables

David A. Hamilton, Jr., and Donna L. R. Wendt

INTERMOUNTAIN FOREST AND RANGE EXPERIMENT STATION
Forest Service
U.S. Department of Agriculture
Roger R. Bay, Director

THE AUTHORS

DAVID A. HAMILTON, JR., is a research forester working in timber measurements and management planning research at the Forestry Sciences Laboratory, Moscow, Idaho. Since receiving his Ph. D. in forestry from Iowa State University in 1970, he has worked on the sampling and modeling of forest mortality.

DONNA L. R. WENDT is a computer scientist now programing for the Data Processing Department at Tacoma Power and Light. After earning a bachelor's degree in mathematics at Washington State University (1972), she spent 15 months as a scientific programer at the Forestry Sciences Laboratory, Moscow, Idaho.

CONTENTS

Page

INTRODUCTION. 1

USES AND PROPERTIES OF SCREEN 3

GUIDES FOR THE USE OF SEARCH AND GRAPH 5

USER INFORMATION FOR GRAPH. 15

SAMPLE JCL FOR THE IBM 360/67 ØS 18

LITERATURE CITED. 20

ABSTRACT

The algorithm reported here is a modeling tool that screens potential relationships between a set of independent variables and a dichotomous dependent variable. Uses of the algorithm and its properties are discussed. A user's guide explains the preparation of input cards for the two PL/1 procedures and explains the program output.

INTRODUCTION

The screening algorithm in the computer program SCREEN was designed to aid in the selection of that set of independent variables that best predicts the outcome of a dichotomous dependent variable. A dichotomous dependent variable is one for which the response is limited to one of two possible outcomes. The algorithm and an example of its use were discussed by Gleser and Collen (1972). The theory behind the algorithm was presented by Sterling and others (1969). SCREEN consists of two PL/1 procedures: SEARCH and GRAPH. SEARCH screens the data for relations between the proportions of the two possible outcomes of the dependent variable and the explanatory independent variables. GRAPH then prints the results of this screening in a format similar to a decision "tree."

Figure 1 is a diagram of a decision "tree." The black squares in the figure are referred to as nodes. At each node, SEARCH determines the most significant independent variable. If the selected independent variable is significant at the user-supplied

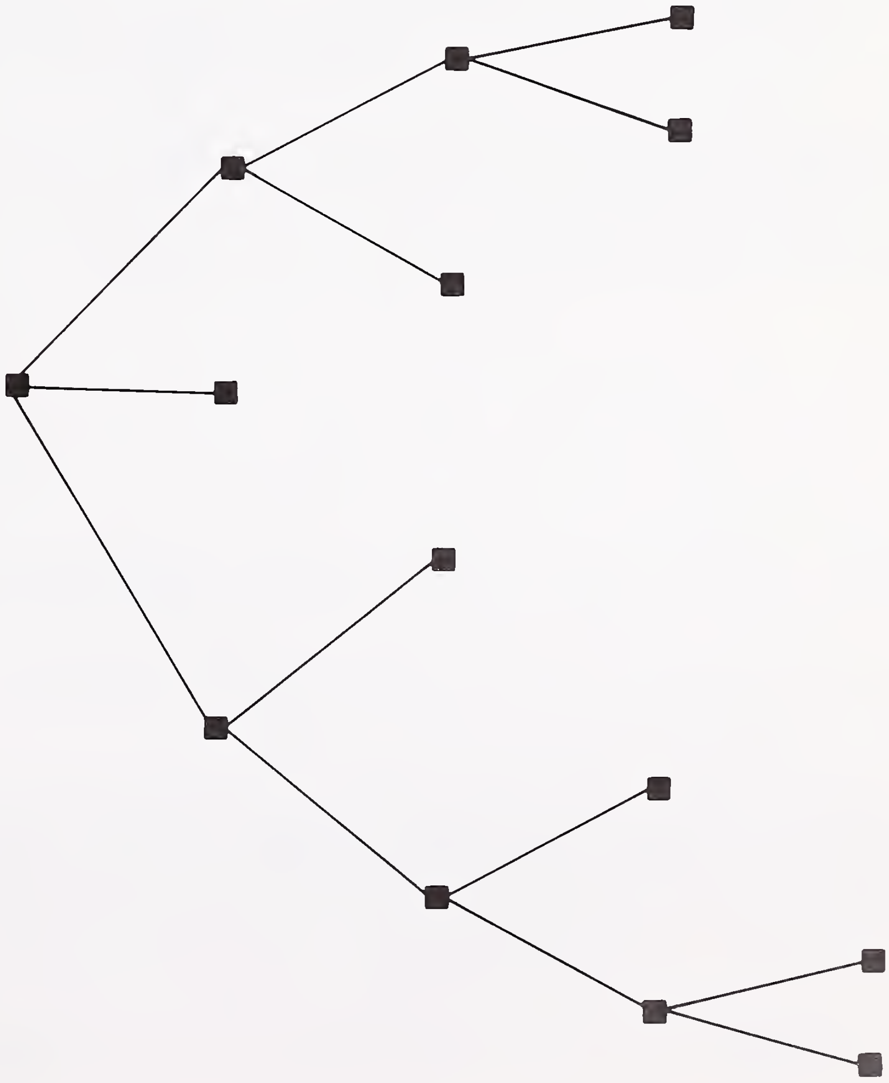


Figure 1. Decision "tree."

```
      PROBABILITY OF MORTALITY (WWPV),95% SIGNIFICANCE LEVEL

                                         TIP VIGOR
                                             POOR
                           CROWN CLASS       FAIR
                               SUP           ........
                               INT           .000044.
                               CODOM         .000061.
                               ........      ........
             CROWN VIGOR       .000075.      GOOD
                 POOR          .000076.      ........
                 ........      ........      .000031.
                 .000084.                    .000015.
                 .000076.                    ........
                 ........
                               DOM
                               ........
                               .000009.
                               .000000.
                               ........

STATUS           FAIR
    ALIVE        ........
    DEAD         .000159.
    ........     .000055.
    .000761.     ........
    .000195.
    ........
                           DBH AS PERCE
                               0-15
                               ........
                               .000046.
                               .000018.
                               ........
                 GOOD
                 EXCELL                  CROWN RATIO
                 ........                    0-2
                 .000518.      15-30         2-4
                 .000064.      30-45         ........
                 ........      45-60         .000115.
                               60-70         .000020.
                               70-80         ........
                               80-90
                               90-100        4-5       DIE BACK
                               ........      5-6           EXTEN
                               .000472.      6-7           ........
                               .000046.      7-8           .000003.
                               ........      8-9           .000003.
                                             9-10          ........
                                             ........      MOD
                                             .000357.      ABSENT
                                             .000026.      ........
                                             ........      .000354.
                                                           .000023.
                                                           ........
```

Figure 2. Output "tree" produced by GRAPH showing results of screening done in SEARCH.

significance level, it is included in the decision "tree" as the next best predictor of the two possible outcomes of the dependent variable. An example of the decision "tree" produced by GRAPH is given in figure 2.

SEARCH and GRAPH were originally written by Malcolm Gleser in PL/1 48 character set[1] and have been compiled under version 5 of the PL/1 compiler. The present authors modified the procedures and prepared this documentation of the modified procedures.

[1] IBM System/360 operating system: PL/1 (F), Language Reference Manual Order No. GC28-8201.

USES AND PROPERTIES OF SCREEN

This method of analysis has a large number of uses, each of which can be classified either as a screening technique, preliminary to model building, or as a modeling technique. In either application, the user must remember that the results of a screen of relationships between dependent and independent variables only provide a portion of the information needed for the development of a model. Known biological relationships and constraints must also be included in the modeling process.

Many screening algorithms assume a specific functional relationship between the dependent and the independent variables. Grosenbaugh (1967) and Furnival (1971) described such screening algorithms. Stepwise-regression algorithms are additional examples of such algorithms, in which screening is accomplished by computing some goodness-of-fit statistic for each set of independent variables to be considered. SEARCH, however, assumes no functional relationship.

When SEARCH is used as a data screening procedure, results provide the user with an optimal set of independent variables for describing the behavior of the dependent variable. If a model is to be constructed from this set of independent variables, the user must determine the nature of the functional relationship between the dependent and independent variables and the validity of any variable transformations that might be expected to improve the goodness-of-fit of the relationship.

Similarly, the user must work with independent variables that are either discrete, with no more than eight classes, or that can be transformed into discrete classes. When continuous variables are transformed, discrete classes need not be of equal width. For example, the variable age might be grouped: age unknown, 0-30 years, 31-40 years, 41-50 years, 51-75 years, and 75 years and over. Discrete classes for continuous independent variables should be defined with considerable care. If classes are too broad, differences in the relationship between dependent and independent variables may be masked. The algorithm will combine classes in which the relationship is similar. Thus, it is usually preferable to define too many discrete classes than to define a few broad classes.

The algorithm described in the SEARCH procedure is independent of most assumptions concerning the numerical structure of data. No distributional assumptions are required for the independent variables other than that they be discrete or readily transformed to discrete variables Screening is not restricted by the need to make any assumptions about the nature of the functional relationship between the dependent and independent variables. Also, screening is unaffected by any transformations of the independent variables as long as a one-to-one relationship is maintained between transformed and untransformed variables.

SEARCH does provide guidance as to significant interactions between independent variables, as is shown by the example of program output in figure 2. For white pine with POOR crown vigor, crown class is the next most significant predictor of mortality. For white pine with GOOD or EXCELLENT crown vigor, d.b.h. as a percentage is more significant than crown class. This result could be explained by a significant interaction between crown class and crown vigor and between d.b.h. as a percent and crown vigor.

We have used SEARCH extensively as a screening technique prior to model building. Currently (1974), we are conducting a study to develop models that will predict tree mortality as a function of anatomical characteristics of the tree. The output "tree" in figure 2 is the output from the GRAPH procedure that resulted from analyzing a population of western white pine that had been measured at 5-year intervals for 20 years (1941-1961). The population is made up of 956 tree records, each of which consists of observations on 15 variables. This population will be used as an example throughout this report to show the data input required to operate SEARCH and GRAPH and to help explain the output of these two procedures. The set of variables selected by SEARCH will be used to develop a functional model that will predict the probability of a white pine becoming a mortality tree in any given 5 years. Similar efforts are underway to develop models that will predict annual mortality rates for all northern Idaho species.

It is not always necessary to develop a functional model from the output of GRAPH. Frequently, the user may be seeking only to identify those variables of importance for future study. In such situations, the "tree" printed by GRAPH provides all information required as to which independent variables are significant predictors of the dependent variable.

This use of SEARCH is more nearly equated with the use for which the procedure was developed (Gleser and Collen 1972). We have used SEARCH to analyze a data set collected to investigate levels and trends of natural inactivation of blister rust cankers on western white pine. The "tree" printed by GRAPH indicates which variables provide most information about the active or inactive status of a canker.

We feel that SEARCH is a valuable data-screening tool in any situation for which the dependent variable is dichotomous. SEARCH could also be used to study:

1. The presence or absence of a resistance mechanism to mountain pine beetle infestation in lodgepole pine;

2. regeneration in order to predict presence or absence of stocked quadrats;

3. the presence or absence of cone serotiny in lodgepole pine; and

4. the presence or absence of cull volume in apparently sound trees.

In each of these examples, SEARCH would be used to select those variables that best predict the relative frequency of occurrence of the two states of the dependent variable.

GUIDES FOR THE USE OF SEARCH AND GRAPH

Three distinct steps are included in an analysis of data by the SCREEN program. First, the data to be analyzed must be transformed and written in a specific format in file INPUT. The actual screening is then performed by executing the PL/1 procedure, SEARCH. Finally, the results of the data screening are printed in a decision "tree" format by executing the PL/1 procedure, GRAPH.

The input for SEARCH is on two files:

File INPUT:

File INPUT must be created in a separate job and written on a data set. This file contains observation records, including both independent and dependent variables.

When the INPUT data set is allocated, LRECL (logical record length) should be set equal to the number of variables; for example, if each record consists of 14 variables, LRECL should be exactly 14 bytes. Each byte on a record represents one variable, that is, byte 1 is for variable 1, byte 2 is for variable 2, and so on. Each independent variable can take on a character value of from 0 to 7, and each dependent variable can take on a character value of 0 or 1. If an independent variable has only two possible outcomes, these outcomes must be represented by 0 or 1. Similarly, if there are only three outcomes, the outcomes are 0, 1, or 2. All records must be complete since missing data would be interpreted as a zero code. A possible solution to the missing data problem is to include an "unknown" class for those variables affected.

Since the tree records for the example population consisted of 15 variables, the record length for INPUT was 15. These variables were as follows:

Variable	No. of outcomes	Possible outcomes
1. DBH	7	(0) 2-10; (1) 10-15; (2) 15-20; (3) 20-25; (4) 25-30; (5) 30-35; (6) 35-40
2. BOLE COND	6	(0) OK; (1) UNSUCCESSFUL BEETLE ATTACK; (2) SUCCESSFUL BEETLE ATTACK; (3) BROKEN TOP; (4) DEAD FORK; (5) ROT
3. CROWN CLASS	4	(0) SUPPRESSED: (1) INTERMEDIATE; (2) CODOMINANT; (3) DOMINANT
4. TIP CHAR	3	(0) BROKEN: (1) SPRAYED OUT; (2) POINTED
5. TIP VIGOR	3	(0) POOR; (1) FAIR; (2) GOOD
6. CROWN WIDTH	3	(0) NARROW; (1) MEDIUM; (2) WIDE

Variable	No. of outcomes	Possible outcomes
7. CROWN RATIO	8	(0) 0-2; (1) 2-4; (2) 4-5; (3) 5-6; (4) 6-7; (5) 7-8; (6) 8-9; (7) 9-10
8. CROWN FORM	5	(0) RAGGED; (1) RAGGED 1-SIDED; (2) 1-SIDED; (3) UNIFORM-RAGGED; (4) UNIFORM
9. CROWN COLOR	3	(0) YELLOW; (1) LIGHT GREEN; (2) GREEN
10. DIE BACK	3	(0) EXTENSIVE; (1) MODERATE; (2) ABSENT
11. CROWN DENSITY	3	(0) POOR; (1) FAIR; (2) GOOD
12. CROWN VIGOR	4	(0) POOR; (1) FAIR; (2) GOOD; (3) EXCELLENT
13. STATUS	2	(0) ALIVE; (1) DEAD
14. DBH AS PERCENT	8	(0) 0-15; (1) 15-30; (2) 30-45; (3) 45-60; (4) 60-70; (5) 70-80; (6) 80-90; (7) 90-100
15. SITE INDEX	8	(0) 0-40; (1) 40-50; (2) 50-60; (3) 60-70; (4) 70-80; (5) 80-90; (6) 90-100; (7) 100+

Suppose columns 1-15 of the first record contained the following values:

2 0 1 2 1 0 4 0 1 0 0 1 1 1 3

Then, the first observation represents a tree with the following characteristics:

(A) Diameter 15-20 inches (2);

(B) OK bole condition (0);

(C) Intermediate crown class (1);

(D) Pointed tip (2);

(E) Fair tip vigor (1);

(F) Narrow crown width (0);

(G) Live crown for 60-70 percent of total height (4):

(H) Ragged crown form (0);

(I) Crown color light green (1):

(J) Extensive dieback (0);

(K) Poor crown density (0);

(L) Fair crown vigor (1);

(M) Dead (1);

(N) Percentage position in the d.b.h. distribution of 15-30 percent (1);

(O) Site index 60-70 (3).

File SYSIN (card input):

Card Type 1

Columns	Variable	Type	PL/1 format	Description
1-6	NVAR	Integer	F(6)	Number of variables
7-12	NLR	Integer	F(6)	Number of observations
13-18	LRECL	Integer	F(6)	Logical record length of file INPUT
24	INPT	Integer	F(6)	= 1 if file INPUT can be held in memory all at once = 0 or blank if file INPUT is to be read in one record at a time

Discussion of Card Type 1: If file INPUT is small enough to be held in memory all at once, then on many computers it is more efficient to do so. In this case, column 24 of card 1 should contain a 1.

However, if the product of NVAR and NLR is greater than 32,676, this method of reading file INPUT results in subscripts being created in SEARCH that exceed the magnitude of subscripts permitted in PL/1. Thus, the file must be read and processed one observation at a time. In this case, column 24 should contain a zero or blank.

A way to reduce input costs when each record is read one at a time is to create file INPUT with a large block size, which reduces the input-output count. However, this method also requires a substantial increase in the amount of memory needed to execute the program. We have found that the computer price structure at Washington State University results in lower costs when the input-output count is reduced at the expense of increased memory sizes. A second means of handling large INPUT files is to screen a subset of the observations. When a subset is screened, care must be taken to assure that the subset is a valid sample of the population of interest.

Card Type 2: The Variable Name Cards

Columns	Variable	Type	PL/1 format	Description
1	NCTG(I)	Integer	F(1)	The number of possible outcomes (2-8) for the Ith variable
2-13	VN(I)	Character	A(12)	The name of the Ith variable
17-24	VNS(I,J)	Character	A(8)	Name of the Jth outcome
25-32 . . .				(J=1,...,8) for the Ith variable

Discussion of Card Type 2: For each variable there must be one card containing the above information. The variable name cards are ordered to correspond to the order of each variable on the input record. There must be a variable name card for each variable on the input record.

Card Type 3: The Variable Inclusion Card(s)

Columns	Variable	Type	PL/1 format	Description
1	INDEX(1)	Integer	A(1)	= blank if variable 1 is to be used as a predictor = 1 if variable 1 is to be omitted or is a dependent variable
2	INDEX(2)	Integer	A(1)	= blank if variable 2 is to be used as a predictor = 1 if variable 2 is to be omitted or is a dependent variable
. . .	. . .	. . .	. . .	. . .
NVAR	INDEX(NVAR)	Integer	A(1)	= blank if variable NVAR is to be used as a predictor = 1 if variable NVAR is to be omitted or is a dependent variable

Discussion of Card Type 3: The variable inclusion control card(s) is used to specify which variables in the input record are to be used as independent variables. Each column of the variable control card corresponds to a position on each of the data input records. If the variable in position 30 of the input record is not to be used as an independent variable, a 1 is entered in column 30 of the variable control card. If there are more than 80 variables per record, several variable inclusion cards are used. Thus, if a record contains 220 variables and if variable number 220 is not to be used as an independent variable, a 1 is placed in column 60 of the third variable inclusion card. Each blank column of the variable control card will cause that variable to be included in the analysis as an independent variable. Any column corresponding to a dependent variable must contain a 1.

Card Type 4: The Control of Analysis Card

Columns	Variable	Type	PL/1 format	Description
1-6	NTOTAL	Integer	F(6)	Number of observations to be used in this analysis, $\leq$NLR
7-12	DVAR#	Integer	F(6)	Column number of the dependent variable on the logical input record

Columns	Variable	Type	PL/1 format	Description
13-18	LEVEL#	Integer	F(6)	Depth to which SEARCH will go, or the maximum number of nodes along each branch. LEVEL# usually is ≤8, but it can be as large as 12
19-24	FLEV#	Integer	F(6)	Level of the first node. This is blank or zero in the normal circumstance. If it is ≥1, then the next card will be card type 5 with information as to what the first nodes will be

Discussion of Card Type 4: NTOTAL can be used to limit the analysis to a subset of the total population. Any value of NTOTAL less than NLR will limit the analysis to the first NTOTAL observations in the population.

LEVEL# is usually specified to be ≤8. Rarely in our uses of SEARCH have as many as 8 variables been significant predictors of the outcome of the dependent variable. In addition, an 8-node "tree" output produced by GRAPH just fits across the width of a page of computer paper.

Card Type 5: The Node Specification Card

Columns	Variable	Type	PL/1 format	Description
1-3	TB(1,1)	Integer	F(3)	Variable number for first forced node
4	TB(1,3)	Integer	F(1)	Lowest desired independent outcome number for first forced node
5	TB(1,4)	Integer	F(1)	Highest desired independent outcome number for first forced node
6-8	TB(2,1)	Integer	F(3)	Variable number for second forced node
9	TB(2,3)	Integer	F(1)	Lowest desired independent outcome number for second forced node
10	TB(2,4)	Integer	F(1)	Highest desired independent outcome number for second forced node
⋮	⋮	⋮	⋮	⋮
(5*I-4)- (5*I-2)	TB(I,1)	Integer	F(3)	Variable number for Ith forced node, where I = FLEV#
(5*I-1)	TB(I,3)	Integer	F(1)	Lowest desired independent outcome number for Ith forced node
(5*I)	TB(I,4)	Integer	F(1)	Highest desired independent outcome number for Ith forced node

Discussion of Card Type 5: This card is included only if the value of FLEV# (forced level number) on card type 4 is ≥1. This card is used to limit the data considered by the SEARCH algorithm to a subset of the population. For the Ith forced variable, three values, TB(I,1), TB(I,3), and TB(I,4), must be read from the node specification card.

In the example discussed previously, we might be concerned with predicting mortality only for those trees with d.b.h. >20 inches and with wide crowns. Thus, FLEV# would be set equal to 2. D.b.h. is variable 1. Outcomes 3, 4, 5, and 6 represent trees >20 inches. Thus, the first 5 columns of the node specification card would contain the values:

00136

Crown width is variable 6. Wide crowns are coded 2. Thus, columns 6 through 10 of the node specification card would contain the values:

00622

This card will cause SEARCH to include in the screening only those observations with d.b.h. >20 inches, coded 3, 4, 5, or 6, and wide crowns coded 2.

Card Type 6: The Significance Level Card

			PL/1	
Columns	Variable	Type	format	Description
1-5	X05(1)	Real	F(5,1)	Chi-square value for 1 degree of freedom at the user supplied significance level
6-10	X05(2)	Real	F(5,1)	Chi-square value for 2 degrees of freedom at the user supplied significance level
⋮	⋮	⋮	⋮	⋮
31-35	X05(7)	Real	F(5,1)	Chi-square value for 7 degrees of freedom at the user supplied signifiance level
36-40	SIGLEV	Real	F(5)	% of the user supplied significance level times 100; for example, a 95% significance level would be expressed as 9500

Discussion of Card Type 6: At each node, SEARCH selects the most significant independent variable only if that variable is significant at the user-supplied significance level. If no independent variable is significant, that branch of the "tree" diagram is terminated. Thus, the extent of the diagram can be controlled by manipulating the significance level.

In certain cases, it is desirable to specify a low significance level such as 50 or 75 percent. If no strong relationship exists between dependent and independent variables, a low significance level will result in independent variables being ranked according to the relative importance. Similarly, a low value of SIGLEV may be necessary if the population being screened is small.

Cards Necessary for Multiple Runs:

SEARCH is designed to permit the use of the same INPUT file for multiple runs of the program. This feature of SEARCH is of value to the user who wishes to screen a data set at a number of significance levels or who wishes to investigate the results of alternative restrictions of the set of independent variables. When multiple runs are desired, the following control cards must be included for each run after the initial run:

1. Variable inclusion card(s);

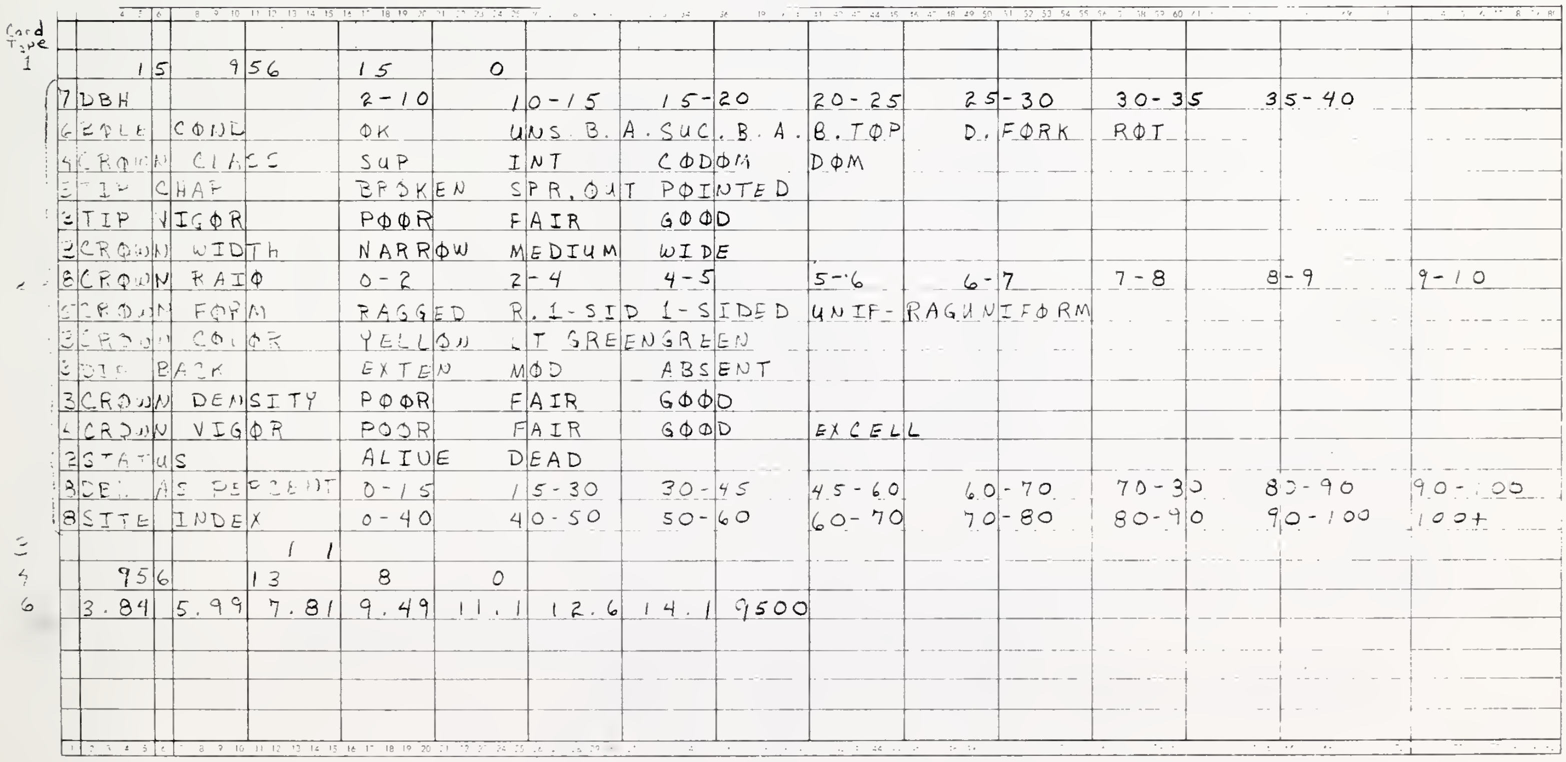
2. the control of analysis card; and

3. the significance level card.

Example of Input Cards for File SYSIN:

Figure 3 provides an example of input cards that were required to run the example discussed previously. Card type 5 is omitted because we did not force any nodes in this example.

Card type 1 tells us that each of the 956 data records in the population is made up of 15 variables; so the record length of file INPUT is set at 15. File INPUT is to be read one record at a time.



```
Card
Type
 1       15   956   15      0
    7DBH           2-10     10-15   15-20   20-25   25-30   30-35   35-40
    6BΦLE CΦND     ΦK       UNS.B.A.SUC.B.A.B.TΦP   D.FΦRK  RΦT
    4CRΦWN CLASS   SUP      INT     CΦDΦM   DΦM
    3TIP CHAR      BRΦKEN   SPR,ΦUT PΦINTED
    3TIP VIGΦR     PΦΦR     FAIR    GΦΦD
    3CRΦWN WIDTH   NARRΦW   MEDIUM  WIDE
 2  8CRΦWN RAIΦ    0-2      2-4     4-5     5-6     6-7     7-8     8-9     9-10
    5CRΦWN FΦRM    RAGGED   R.1-SID 1-SIDED UNIF-RAGUNIFΦRM
    3CRΦWN CΦLΦR   YELLΦW   LT GREENGREEN
    3DIE BACK      EXTEN    MΦD     ABSENT
    3CRΦWN DENSITY PΦΦR     FAIR    GΦΦD
    4CRΦWN VIGΦR   POOR     FAIR    GΦΦD    EXCELL
    2STATUS        ALIVE    DEAD
    8DEF AS PERCENT0-15     15-30   30-45   45-60   60-70   70-80   80-90   90-100
    8SITE INDEX    0-40     40-50   50-60   60-70   70-80   80-90   90-100  100+
 3              1 1
 4     956     13    8     0
 6   3.84 5.99 7.81 9.49 11.1 12.6 14.1 9500
```

Figure 3. Example of input cards needed to operate SEARCH.

The set of type 2 cards defines the 15 variables. The number and the definition of the classes of each variable are also specified on these cards.

Card type 3 excludes variables 13 (status) and 15 (site index) from the set of independent variables.

Card type 4 states that 956 observations will be used in this analysis. Variable 13 will be the dependent variable for this run. SEARCH will run for a maximum of 8 nodes. No nodes will be forced in this run.

Since no nodes are to be forced, card type 5 is unnecessary. Card type 6 specifies that the significance level for this run will be 95 percent. The first seven numbers on this card are the significant chi-square values for 1 through 7 degrees of freedom.

OUTPUT From SEARCH

Output consists primarily of punched cards to be used in a second procedure, GRAPH. However, some printed output is provided.

The first page of printed output contains two blocks of printout. The first block consists of the variable name cards that correspond to those variables to be used as independent variables. The second block contains the variable name cards that correspond to those variables not to be used as independent variables.

Values of the following variables are printed on the second page:

Variable	Description
NTOTAL	Same as NTOTAL on control of analysis card.
NVAR	Same as NVAR on Card Type 1
DVAR#	Same as DVAR# on control of analysis card
LRECL	Same as LRECL on Card Type 1
INPT	Same as INPT on Card Type 1
LEVEL#	Same as LEVEL# on control of analysis card
FLEV#	Same as FLEV# on control of analysis card
MNUM	Minimum number of observations required in each category
X05(1)-X05(7), SIGLEV	Same as on significance level card
TAB(I,J)	A series of 2x N_K contingency tables formed by cross classifying the population by the dependent and by each independent variable. J can be 0 or 1, and I runs from 1 to $\sum_{K=1}^{INDPT} N_K$, where

Variable Description

INDPT = number of included independent variables used in the run and N_K = number of possible outcomes for the Kth included independent variable.

The dependent variable in the example discussed earlier is STATUS and the first independent variable is DBH with 7 possible outcomes. Thus TAB(I,J) is defined as:

TAB(1,0) = number of observations with "0" STATUS in in the "0" DBH category

TAB(1,1) = number of observations with "1" STATUS in the "0" DBH category

TAB(2,0) = number of observations with "0" STATUS in the "1" DBH category

TAB(2,1) = number of observations with "1" STATUS in the "1" DBH category

.
.
.

TAB(7,0) = number of observations with "0" STATUS in the "6" DBH category

TAB(7,1) = number of observations with "1" STATUS in the "6" DBH category

The next independent variable is BOLE COND with 6 possible outcomes. Thus:

TAB(8,0) = number of observations with "0" STATUS in the "0" BOLE COND category

TAB(8,1) = number of observations with "1" STATUS in the "0" BOLE COND category

.
.
.

TAB(13,1) = number of observations with "1" STATUS in the "5" BOLE COND category

The final independent variable is SITE INDEX with 8 possible outcomes. Thus, the final element of TAB(I,J) is:

TAB(68,1) = number of observations with "1" STATUS in the "7" SITE INDEX category

DØUT(0) = total number of observations with a dependent variable of 0.

DØUT(1) = total number of observations with a dependent variable of 1.

NOTE: DØUT(0) + DØUT(1) should equal NTOTAL

The punched card output produced by SEARCH is also printed as the next lines of output. The final line of output lists the value of EXIT#. The program terminates by encountering an end of file on file SYSIN or on file INPUT. EXIT# identifies the particular GET statement where SEARCH terminates.

The following chart lists the meaning of the different EXIT#'s:

EXIT #	Procedure card sequence number where end of file occurred	Condition of termination
1	SRCH 190	End of file encountered as program read Card Type 1
2	SRCH 550	Normal termination
3	SRCH 580	NVAR>80. End of file encountered as program read continuation of variable inclusion card
4	SRCH 450	End of file encountered as program read variable name card
5	SRCH 990	End of file encountered as program read control of analysis card
6	SRCH1040	End of file encountered as program read node specification card
8	SRCH1530	End of file encountered as program read significance level card
11	SRCH3060	End of file encountered as program read file INPUT one observation at a time
12	SRCH3000	End of file encountered as program read file INPUT all at one time

Occurrence of EXIT#'s 1,3,4,5,6, or 8 indicates that the input cards to file SYSIN have been prepared incorrectly or that some have been omitted. Occurrence of EXIT#'s 11 or 12 indicates that there are fewer observations in file INPUT than were recorded on card type 1.

SEARCH also creates a temporary file PASS. This file contains the same information as the punched card output records created by SEARCH, which permits the user to run SEARCH and GRAPH as two job steps in a single batch job. The job-control cards needed to do this are listed in the SAMPLE JCL (job-control language) section following the discussion of GRAPH.

USER INFORMATION FOR GRAPH

GRAPH is the procedure that prints the "tree" diagram(s) generated by the SEARCH procedure.

SYSIN and PASS are the two input files for GRAPH. SYSIN is card input prepared by the user. PASS either can be the punched cards produced by SEARCH or, if SEARCH and GRAPH are run as two job steps in the same batch job, a temporary data set passed from SEARCH. Details of the job-control language needed to use these files are presented in the SAMPLE JCL section.

File SYSIN must contain the following cards:

Card Type 1

Columns	Variable	Type	PL/1 format	Description
1-6	NVAR	Integer	F(6)	Number of variables

Card Type 2

These are the same as the variable name cards used in SEARCH, where

N = number of categories

VN(I) = name of the Ith variable

VNS(I,J) = name of the Jth category for the Ith variable

Card Type 3

Columns	Variable	Type	PL/1 format	Description
1-3	NNN	Integer	F(3)	Position number of the dependent variable, same as DVAR# in SEARCH
4-80	TITLE	Character	A(77)	Title to be used as page header for the "tree" diagram

Card Type 4: Control of Analysis Card

Columns	Variable	Type	PL/1 format	Description
19-24	FLEV#	Integer	F(6)	Number of forced nodes in this run

The value for FLEV# on this card must be the same as the value for FLEV# used in SEARCH.

Cards Necessary for Multiple Runs:

Multiple runs of GRAPH are also possible. For each "tree" diagram that is to be printed after the initial diagram, the following control cards must be included:

In file SYSIN,

1. Title card; and
2. control of analysis card.

In file PASS,

1. Punched card output or equivalent records in temporary data set.

Example of Input Cards for File Sysin:

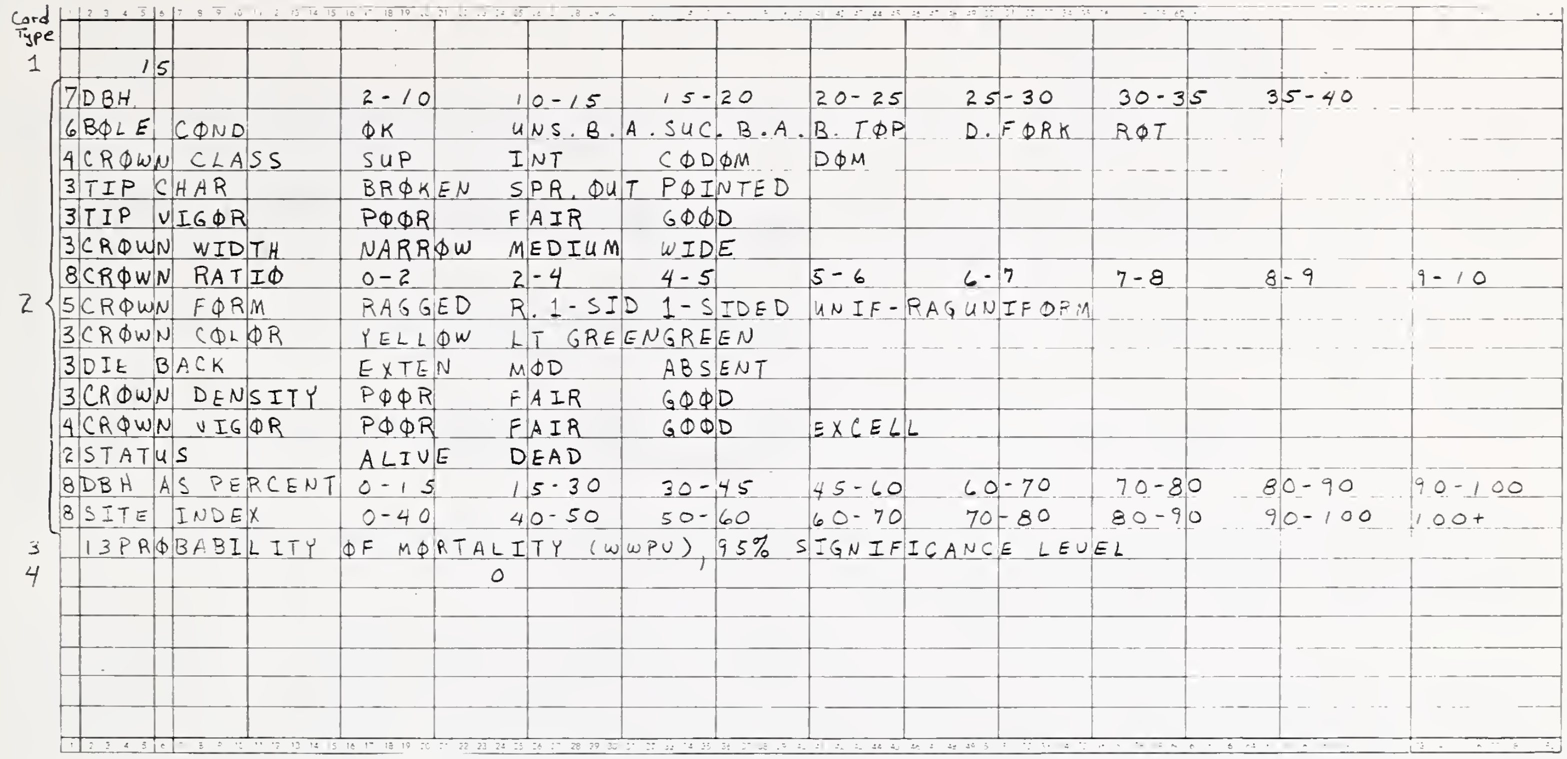
Figure 4 provides an example of the input cards needed to print the "tree" diagram produced by the SEARCH run described earlier.

Printed "Tree" Diagram Output From Graph

Printed for each problem run in SEARCH is a "tree" diagram, consisting of the dependent variable as a root and combinations of significant predictors as branches. The "tree" should be read from left to right. Each branch represents the best set of predictors for a specific combination of independent variable outcomes.

The sample "tree" printout in figure 2 reports the optimal set of predictors for western white pine mortality. This "tree" was produced by the SEARCH run described in figure 3 and by the GRAPH run described in figure 4. Crown vigor categories separate white pine into three different mortality ratio classes. White pine with either good or excellent crown vigor apparently have similar mortality ratios. Within the poor crown-vigor category, the next best predictor is crown class. White pines with codominant, intermediate, or suppressed crown class have similar mortality ratios, which is significantly different from the mortality ratio associated with white pine with dominant crown class. By contrast, for those white pines with good or excellent crown vigor, d.b.h. as a percent is the next best predictor, although only the lowest class has a mortality ratio different from the other classes. For white pine with fair crown vigor, none of the potential independent variables provides any further significant discriminating power.

For white pine with poor crown vigor and codominant, intermediate, or suppressed crown class, tip vigor is the next best predictor. Trees with good tip vigor have a



```
Card
Type
 1      15
    7DBH            2-10    10-15   15-20   20-25   25-30   30-35   35-40
    6BØLE CØND      ØK      UNS.B.A.SUC.B.A.B.TØP   D.FØRK  RØT
    4CRØWN CLASS    SUP     INT     CØDØM   DØM
    3TIP CHAR       BRØKEN  SPR.ØUT PØINTED
    3TIP VIGØR      PØØR    FAIR    GØØD
    3CRØWN WIDTH    NARRØW  MEDIUM  WIDE
    8CRØWN RATIØ    0-2     2-4     4-5     5-6     6-7     7-8     8-9     9-10
 2  5CRØWN FØRM     RAGGED  R.1-SID 1-SIDED UNIF-RAGUNIFØRM
    3CRØWN CØLØR    YELLØW  LT GREENGREEN
    3DIE BACK       EXTEN   MØD     ABSENT
    3CRØWN DENSITY  PØØR    FAIR    GØØD
    4CRØWN VIGØR    PØØR    FAIR    GØØD    EXCELL
    2STATUS         ALIVE   DEAD
    8DBH AS PERCENT 0-15    15-30   30-45   45-60   60-70   70-80   80-90   90-100
    8SITE INDEX     0-40    40-50   50-60   60-70   70-80   80-90   90-100  100+
 3   13PRØBABILITY  ØF MØRTALITY (WWPU), 95% SIGNIFICANCE LEVEL
 4                         0
```

Figure 4. Example of input cards needed to operate GRAPH.

different mortality ratio than do trees with poor or fair tip vigor. For those trees with poor crown vigor and dominant crown class, there are no further significant predictors. For those trees with good or excellent crown vigor and d.b.h. as a percent greater than 15 percent, crown ratio and dieback also are significant predictors.

The numbers in the boxes have a specific meaning. The box under GOOD tip vigor contains a 31 and a 15. The 31 indicates that there were 31 live white pines that had characteristics of POOR crown vigor, that were in the SUP-INT-CODOM crown class, and that had GOOD tip vigor. The 15 indicates that there were 15 dead white pines with the same set of characteristics.

At the completion of all "tree" printouts, EXIT# is listed to indicate the point of termination in GRAPH.

The following chart lists the meaning of the different EXIT#'s:

EXIT#	Procedure card sequence number where end of file occurred	Condition of termination
1	GRPH 60	End of file encountered as program read Card Type 1
2	GRPH 400	Normal termination

EXIT#	Procedure card sequence number where end of file occurred	Condition of termination
3	GRPH 590	End of file encountered as program read file PASS
4	GRPH 540	End of file encountered as program read control of analysis card
5	GRPH 310	End of file encountered as program read the variable name cards

The occurrence of EXIT#'s 1,4, or 5 indicates that file SYSIN has been prepared incorrectly or that some needed cards have been omitted. Occurrence of EXIT# 3 indicates that file PASS is incomplete.

The following examples of JCL are provided for the guidance of those who will run the program on a standard IBM 360 Operating System. To those who use computers with other operating systems, examples show files and parameters that must be defined for the operation of SEARCH and GRAPH.

SAMPLE JCL FOR THE IBM 360/67 ØS

To run SEARCH alone from the deck:

```
JØB CARD
// EXEC PL1LFCLG,PARM.PL1L='C48',REGIØN.PL1L=98K,
// REGIØN.GØ=130K
//PL1L.SYSIN DD *

     SEARCH deck

//GØ.SYSPRINT DD SYSØUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//GØ.PUNCH DD SYSØUT=B
//GØ.PASS DD DSN=&&TEMP,DISP=(NEW,PASS),SPACE=(TRK,(4,1)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7280,DSØRG=PS),UNIT=SYSSCR
//GØ.INPUT DD DSN=data set name of observation data set created in previous job,
     DISP=SHR
//GØ.SYSIN DD *

     Data input cards
```

To run GRAPH alone from the deck:

```
JØB CARD
// EXEC PL1LFCLG,PARM.PL1L='C48',REGIØN.PL1L=98K,
// REGIØN.GØ=138K
//PL1L.SYSIN DD *
```

```
     GRAPH deck

//GØ.SYSPRINT DD SYSØUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//GØ.PASS DD *
     punched cards from SEARCH
//GØ.SYSIN DD *
     SYSIN cards for GRAPH
```

To run SEARCH and GRAPH together from decks:

```
JØB CARD
// EXEC PL1LFCLG,PARM.PL1L='C48',REGIØN.PL1L=98K,
// REGIØN.GØ=130K
//PL1L.SYSIN DD *

     SEARCH deck

//GØ.SYSPRINT DD SYSØUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//GØ.PUNCH DD SYSØUT=B
//GØ.PASS DD DSN=&&TEMP,DISP=(NEW,PASS),SPACE=(TRK,(4,1)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7280,DSØRG=PS),UNIT=SYSSCR
//GØ.INPUT DD DSN=data set name of observation data set created in previous job,
     DISP=SHR
//GØ.SYSIN DD *

     Data input cards

// EXEC PL1LFCLG,PARM.PL1L='C48',REGIØN.PL1L=98K,
// REGIØN.GØ=138K
//PL1L.SYSIN DD *

     GRAPH deck

//LKED.SYSLMØD DD DSN=&&GØSET(GØ),DISP=(NEW,PASS),
// DCB=BLKSIZE=1024,UNIT=SYSSCR,SPACE=(CYL,(1,1,1))
//GØ.SYSPRINT DD SYSØUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//GØ.PASS DD DSN=&&TEMP,DISP=(ØLD,PASS)
//GØ.SYSIN DD *

     SYSIN cards for GRAPH
```

SEARCH requires 98K bytes of memory to compile. The amount of memory required to execute SEARCH varies from about 100K bytes to 160K bytes depending on the size of the population to be screened and on the number of variables in each observation. GRAPH requires 98K bytes of memory to compile and 138K bytes of memory to execute.

GRAPH will usually execute in less than 1 minute central processing unit (CPU) time. SEARCH will usually execute in less than 2 minutes CPU time. However, an increase in population size, an increase in the number of variables per observation, or a decrease in significance level will result in an increase in the execution time required by SEARCH.

Requests for the program should be directed to:

Intermountain Forest and Range Experiment Station
Forestry Sciences Laboratory
Attention: David A. Hamilton, Jr.
1221 South Main Street
Moscow, Idaho 83843

LITERATURE CITED

Furnival, George M.
1971. All possible regressions with less computation. Technometrics 13(2):403-408.

Gleser, Malcolm A., and Morris F. Collen.
1972. Towards automated medical decisions. Comput. and Biomed. Res. 5:180-189.

Grosenbaugh, L. R.
1967. REX-Fortran-4 system for combinatorial analysis of multivariate regression. USDA For. Serv. Res. Pap. PSW-44.

Sterling, Theodor D., Randall G. Binks, Shelby Haberman, and Seymour V. Pollack.
1969. Robot data screening--a ubiquitous automatic search technique. P. 319-333, in: Milton, Roy C., and John A. Nelder (ed), Statistical Computation. 462 p. Academic Press, New York.

☆U.S. GOVERNMENT PRINTING OFFICE: 1975 — 677-328 / 3 REGION NO. 8

HAMILTON, DAVID A., JR., and DONNA L. R. WENDT
1975. SCREEN: a computer program to identify predictors of dichotomous dependent variables. USDA For. Serv. Gen. Tech. Rep. INT-22, 20 p. (Intermountain Forest & Range Experiment Station, Ogden, Utah 84401.)

The algorithm reported here is a modeling tool that screens potential relationships between a set of independent variables and a dichotomous dependent variable. Uses of the algorithm and its properties are discussed. A user's guide explains the preparation of input cards for the two PL/1 procedures and explains the program output.

OXFORD: U681.3;--015.5.
KEYWORDS: computer programs, statistical methods, data screening algorithm, nonlinear regression, dichotomous dependent variable, modeling.

HAMILTON, DAVID A., JR., and DONNA L. R. WENDT
1975. SCREEN: a computer program to identify predictors of dichotomous dependent variables. USDA For. Serv. Gen. Tech. Rep. INT-22, 20 p. (Intermountain Forest & Range Experiment Station, Ogden, Utah 84401.)

The algorithm reported here is a modeling tool that screens potential relationships between a set of independent variables and a dichotomous dependent variable. Uses of the algorithm and its properties are discussed. A user's guide explains the preparation of input cards for the two PL/1 procedures and explains the program output.

OXFORD: U681.3;--015.5.
KEYWORDS: computer programs, statistical methods, data screening algorithm, nonlinear regression, dichotomous dependent variable, modeling.

HAMILTON, DAVID A., JR., and DONNA L. R. WENDT
1975. SCREEN: a computer program to identify predictors of dichotomous dependent variables. USDA For. Serv. Gen. Tech. Rep. INT-22, 20 p. (Intermountain Forest & Range Experiment Station, Ogden, Utah 84401.)

The algorithm reported here is a modeling tool that screens potential relationships between a set of independent variables and a dichotomous dependent variable. Uses of the algorithm and its properties are discussed. A user's guide explains the preparation of input cards for the two PL/1 procedures and explains the program output.

OXFORD: U681.3;--015.5.
KEYWORDS: computer programs, statistical methods, data screening algorithm, nonlinear regression, dichotomous dependent variable, modeling.

HAMILTON, DAVID A., JR., and DONNA L. R. WENDT
1975. SCREEN: a computer program to identify predictors of dichotomous dependent variables. USDA For. Serv. Gen. Tech. Rep. INT-22, 20 p. (Intermountain Forest & Range Experiment Station, Ogden, Utah 84401.)

The algorithm reported here is a modeling tool that screens potential relationships between a set of independent variables and a dichotomous dependent variable. Uses of the algorithm and its properties are discussed. A user's guide explains the preparation of input cards for the two PL/1 procedures and explains the program output.

OXFORD: U681.3;--015.5.
KEYWORDS: computer programs, statistical methods, data screening algorithm, nonlinear regression, dichotomous dependent variable, modeling.

Headquarters for the Intermountain Forest and Range Experiment Station are in Ogden, Utah. Field Research Work Units are maintained in:

Boise, Idaho
Bozeman, Montana (in cooperation with Montana State University)
Logan, Utah (in cooperation with Utah State University)
Missoula, Montana (in cooperation with University of Montana)
Moscow, Idaho (in cooperation with the University of Idaho)
Provo, Utah (in cooperation with Brigham Young University)
Reno, Nevada (in cooperation with the University of Nevada)

AMERICAN REVOLUTION BICENTENNIAL
1776-1976

GROWING
FROM '76
U.S. FORESTRY
CENTENNIAL